

BAB II

KAJIAN PUSTAKA

A. Kajian Teoritis

1. Rancang Bangun

Rancang bangun adalah serangkaian kegiatan untuk merancang dan mengembangkan suatu sistem melalui tahapan yang terstruktur sehingga menghasilkan program atau aplikasi yang sesuai dengan kebutuhan. Selain itu, rancang bangun mencakup proses perancangan sistem dengan mengimplementasikan hasil analisis dalam perangkat lunak. Tahap ini bertujuan menghasilkan sistem baru atau mengembangkan sistem yang telah ada agar lebih sesuai dengan kebutuhan pengguna.

Menurut Setiawan et al., (2024) mengatakan bahwa rancang bangun merupakan suatu rangkaian proses yang bertujuan untuk menciptakan sistem baru atau meningkatkan sistem yang ada dengan mengubah hasil analisis menjadi bahasa pemrograman.

Rancang bangun merupakan tahap awal dalam proses perancangan yang diawali dengan pembuatan gambaran atau sketsa baru yang sebelumnya belum pernah dibuat. Sketsa tersebut kemudian dikelola dan dikembangkan lebih lanjut menjadi suatu bentuk rancangan yang memiliki fungsi sesuai dengan kebutuhan yang diinginkan. Dalam prosesnya, rancang bangun tidak hanya berfokus pada pembuatan konsep awal, tetapi juga mencakup penyempurnaan desain agar dapat dilaksanakan secara efektif dan efisien. Oleh karena itu, tahapan ini menjadi bagian penting dalam menghasilkan

suatu sistem atau produk yang terstruktur dan memiliki nilai guna yang jelas (Ariansyah & Wijaya, 2021).

Sudut pandang ini mengarah pada kesimpulan bahwa rancang bangun merupakan suatu tahapan awal dalam proses pengembangan sistem yang mencakup penyusunan konsep, pembuatan sketsa, serta penerjemahan hasil analisis ke dalam bentuk yang lebih konkret, seperti bahasa pemrograman atau desain sistem.

2. Sistem Informasi

Sistem informasi merupakan suatu unit yang terdiri dari orang, teknologi, dan proses yang bekerja sama untuk mengubah data menjadi informasi yang bermanfaat dan memfasilitasi pelaksanaan operasi organisasi. Menurut Suli & Nirsal (2023) menjelaskan bahwa Sistem informasi adalah sistem yang mencakup pengumpulan, pemasukan, pengolahan, penyimpanan, pengendalian, dan pelaporan data untuk menyediakan informasi yang membantu suatu organisasi dalam mengambil keputusan dan mencapai tujuan serta sasarannya. Selain itu, setiap sistem informasi memiliki tujuan dan fungsi yang beragam, yang dipengaruhi oleh bagaimana sistem tersebut dirancang, digunakan, serta siapa yang mengoperasikannya (Effendi et al., 2023).

Menurut Dinka et al., (2022), Sistem informasi adalah kumpulan bagian dan operasi yang saling terhubung yang digunakan untuk mengumpulkan, memproses, menyimpan, dan mendistribusikan data menjadi

informasi yang dapat digunakan untuk membantu manajemen dan pengambilan keputusan di dalam suatu bisnis atau organisasi.

Sistem informasi dibangun melalui keterkaitan berbagai unsur yaitu teknologi dan sumber daya manusia dalam suatu organisasi untuk mendukung operasional, termasuk transaksi dan komunikasi. Informasi yang dihasilkan dari sistem ini dapat dimanfaatkan oleh pihak yang membutuhkan secara efektif (Arribe & Ryandi, 2023).

Berdasarkan uraian diatas, sistem informasi adlaah sistem yang mengintegrasikan data, praktik kerja, teknologi informasi, dan sumber daya manusia untuk mengubah data menjadi informasi yang bermanfaat. Pengambilan keputusan dan operasi operasional didukung oleh data ini.

3. Absensi

Absensi atau kehadiran merupakan suatu proses pencatatan yang digunakan untuk mengetahui tingkat partisipasi atau kehadiran individu dalam suatu kegiatan. Dalam konteks organisasi atau perusahaan, absensi digunakan untuk memonitor kehadiran karyawan secara sistematis sebagai bagian (Prayogge & Megawati, 2023). Menurut Shofi et al., (2021), absensi karyawan ditempat kerja dilacak dan dicatat menggunakan sistem absensi sesuai dengan jadwal kerja yang relevan. Pencatatan absensi ini berfungsi sebagai alat kontrol dalam menilai kedisiplinan serta tanggung jawab karyawan dalam menajalankan pekerjaannya.

Maulana et al., (2024) menjelaskan bahwa Absensi atau catatan kehadiran merupakan Catatan waktu kedatangan setiap karyawan selama

mereka bekerja di perusahaan tersebut. Informasi ini digunakan untuk melacak tindakan disiplin dan kehadiran karyawan. Hal ini menunjukkan bahwa data absensi memiliki peran penting dalam memastikan proses penggajian berjalan secara akurat, karena perhitungan kompensasi karyawan dapat disesuaikan dengan riwayat kehadiran dan ketentuan kerja yang berlaku.

Absensi merupakan mekanisme pencatatan kehadiran yang dilakukan untuk mencatat dan memantau kehadiran karyawan dalam suatu kegiatan atau pekerjaan. Dalam lingkungan organisasi atau perusahaan, absensi berfungsi sebagai sarana untuk mengetahui tingkat kehadiran karyawan serta sebagai alat evaluasi dalam menilai kedisiplinan dan kepatuhan terhadap peraturan kerja.

4. Penggajian

Penggajian merupakan kewajiban perusahaan dalam memberikan kompensasi yang diberikan perusahaan kepada karyawan, yang biasanya diberikan secara berkala, seperti setiap bulan, sesuai dengan ketentuan yang berlaku (Dinka et al., 2022). Selain itu, Aminudin & Saripudin (2021), mengatakan bahwa *payroll system there are six procedures, namely the process of producing employee attendance lists, making employee payroll lists, making wage slips, making cash out receipts, making payroll payments, and making payroll reports*. Dijelaskan bahwa pembuatan daftar kehadiran karyawan, daftar gaji karyawan, slip gaji, bukti pengeluaran kas, pembayaran gaji, dan laporan penggajian adalah enam proses utama yang dicakup oleh

sistem penggajian. Keenam langkah ini bekerja sama untuk mengelola data penggajian dan kehadiran, yang akhirnya menghasilkan data upah yang dibutuhkan perusahaan.

Berdasarkan pendapat diatas, dapat disimpulkan bahwa penggajian merupakan proses pemberian kompensasi kepada karyawan atas pekerjaan yang telah dilakukan berdasarkan persyaratan tertentu, seperti data kehadiran, jabatan dan komponen perhitungan gaji lainnya. Sistem penggajian terkomputerisasi penting untuk memungkinkan perhitungan, pembayaran, penyimpanan data, dan pelaporan penggajian yang lebih cepat, lebih tepat, akurat, dan efisien.

5. Website

Website merupakan sekumpulan halaman *website* yang saling terhubung yang digunakan untuk mengkomunikasikan berbagai jenis informasi, termasuk teks, grafik, dan elemen lain yang dapat diakses melalui internet (Suli & Nirsal, 2023). Sedangkan menurut Hendra & Riti (2023) mengatakan bahwa *website* merupakan sekumpulan halaman yang saling terhubung dan digunakan untuk menyajikan informasi mengenai suatu topik tertentu, baik yang berkaitan dengan perusahaan, organisasi, maupun institusi.

Menurut Hasibuan et al., (2023) menjelaskan bahwa *website* merupakan sekumpulan halaman *website* yang saling terhubung dan umumnya berada dalam satu domain atau subdomain yang sama, serta dapat diakses melalui jaringan *Word Wide Web* (WWW) di internet.

Website merupakan sekumpulan halaman *website* yang saling terhubung dalam suatu domain atau subdomain tertentu yang berfungsi untuk menyajikan informasi kepada pengguna. *Website* dapat diakses melalui internet dan dimanfaatkan untuk berbagai keperluan, seperti penyampaian informasi yang berkaitan dengan individu, kelompok, maupun institusi.

a. *Laravel*

“Laravel is a helpful PHP framework that makes it simple to construct online applications. It is built on the Symfony system and employs the well-known MVC (Model-View-Controller) architectural paradigm.” Dijelaskan bahwa *Laravel* memudahkan perancangan aplikasi sesuai kebutuhan. Selain itu, *Laravel* terdapat berbagai fitur, seperti dukungan untuk berbagai sistem basis data, routing yang sederhana, dan kemampuan untuk membangun aplikasi yang aman, mudah dikembangkan, dan lebih efisien (Subecz, 2021).

Siswanto (2023:12), menjelaskan bahwa *Laravel* merupakan *framework* aplikasi *website* berbasis PHP yang digunakan untuk mendukung proses pengembangan aplikasi *website* secara lebih mudah dan terstruktur. Hal ini karena *Laravel* menyediakan fitur yang lengkap serta menerapkan struktur pengembangan yang terorganisir, sehingga dapat membantu pengembang dalam membangun aplikasi secara lebih cepat, efisien, dan mudah dikelola.

Berdasarkan beberapa sudut pandang yang disebutkan diatas, dapat dikatakan bahwa *Laravel* adalah *framework* PHP yang

menggunakan desain *Model View Controller* (MVC) untuk memfasilitasi pengembangan aplikasi *website* yang lebih sistematis. Aplikasi menjadi lebih terorganisir, efektif, dan mudah dipelihara sebagai konsekuensi dari berbagai kemampuan yang membantu dalam pengelolaan data dan antarmuka.

b. PHP

Bahasa pemrograman sisi server yang disebut PHP (*Hypertext Preprocessor*) digunakan untuk membuat situs *website* dinamis. Sebelum membuat HTML yang dilihat pengguna, semua pemrosesan data dilakukan deserver. PHP diciptakan dengan memodifikasi ide-ide dari bahasa pemrograman C, Java, dan *Perl*, serta bersifat *open source* (Zulfa, 2024). Adapun pendapat dari Rompas et al., (2023) mengatakan bahwa PHP adalah bahasa pemrograman sisi server yang dapat dipadukan dengan HTML dan digunakan dalam pembuatan aplikasi online. Kode PHP awalnya dijalankan deserver ketika pengguna mengunjungi situs *website*, setelah itu browser menerima hasil yang telah diproses dan menampilkannya sebagai halaman *website*. Menurut Arisantoso et al., (2024:99), bahwa “PHP adalah bahasa pemrograman *website* sisi server yang digunakan untuk membuat aplikasi online dinamis.”

Berdasarkan pendapat tersebut, dapat disimpulkan bahwa PHP merupakan bahasa pemrograman sisi *server* yang digunakan untuk mengembangkan aplikasi *website* dinamis dengan memproses data di server sebelum ditampilkan kepada pengguna.

c. *Visual Studio Code (VSC)*

Microsoft menciptakan *Visual Studio Code (VSC)*, sebuah editor kode yang kompatibel dengan *Windows*, *Linux*, dan *macOS*, serta operasi lainnya (Ningsih et al., 2022). Selain bersifat *open source* dan gratis untuk digunakan, *Visual Studio Code (VSC)* menawarkan sejumlah fitur yang memudahkan pengembang untuk menulis, memodifikasi, dan mengelola kode program (Hidayah & Rofiqoh, 2024).

Berdasarkan pendapat diatas, dapat disimpulkan bahwa *Visual Studio Code* merupakan editor kode sumber yang dikembangkan oleh *Microsoft*, bersifat ringan, fleksibel, dan dapat digunakan diberbagai sistem operasi. Serta mendukung berbagai bahasa pemrograman melalui fitur bawaan maupun ekstensi.

d. *Bootstrap*

Bootstrap merupakan *framework* yang digunakan untuk mengembangkan tampilan dan tata letak secara fleksibel dengan memanfaatkan komponen CSS dan HTML, sehingga proses pembuatan antarmuka dapat dilakukan dengan lebih cepat dan mudah (Lubis et al., 2022). Sedangkan itu, Putro et al., (2024:120) mengatakan bahwa *Bootstrap* adalah kerangka kerja CSS sumber terbuka yang memudahkan pembuatan antarmuka *website* responsif. Dengan bantuan banyak elemen siap pakai *Bootstrap*, yang meliputi tombol, formular, navigasi, dan sistem grid, pengembang dapat dengan cepat dan efektif membangun tampilan

situs *website* yang menarik secara visual dan konsisten yang dapat menyesuaikan dengan berbagai ukuran layar.

Menurut Dirgantara & Andrian (2023), *Bootstrap* merupakan *framework* CSS yang berisi kumpulan komponen antarmuka, seperti *grid system*, tipografi, *form*, dan tombol, yang dirancang untuk mendukung pembuatan tampilan *website* responsif.

Bootstrap merupakan *framework* CSS yang menyediakan berbagai komponen antarmuka untuk membangun tampilan *website* yang responsif, fleksibel, dan terstruktur secara efisien.

6. Basis Data

Basis data merupakan kumpulan data yang tersusun secara sistematis dan disimpan dalam komputer, sehingga dapat diolah dan dimanfaatkan untuk menghasilkan informasi melalui program komputer (Widodo, 2024). Selain itu, Jamaludin et al., (2022:2) mengatakan bahwa basis data merupakan kumpulan data yang saling terhubung dan tersusun secara sistematis, yang biasanya diorganisasikan menggunakan kunci tertentu untuk memudahkan pengelolaan dan pencarian data.

Menurut Hardiansyah & Dewi (2020), basis data merupakan kumpulan data yang terintegrasi dan dapat dikembangkan untuk memenuhi kebutuhan informasi pengguna dalam suatu organisasi atau perusahaan. Hal ini dapat disimpulkan bahwa basis data merupakan kumpulan data yang tersusun secara sistematis, saling terhubung, dan terintegrasi dalam suatu

sistem komputer, yang dikelola untuk menghasilkan informasi guna memenuhi kebutuhan pengguna.

a. *MySQL*

MySQL merupakan salah satu sistem manajemen basis data yang paling terkenal dan banyak digunakan di seluruh dunia, sebuah program server basis data sumber terbuka (Yunitarini et al., 2023). Sejalan dengan itu, Wahyudi et al., (2022) *“MySQL is also a part of the Relational Database Management System (RDBMS), which has a database structure that allows the program to communicate with the database server when the data retrieval procedure employs the relational database approach.”* Dijelaskan bahwa MySQL memfasilitasi penyimpanan dan pengambilan data relasional dan merupakan bagian dari sistem manajemen basis data relasional (RDBMS). Kerangka kerja ini memfasilitasi komunikasi antara server basis data dan aplikasi, memungkinkan penanganan informasi yang lebih terorganisir dan efisien.

Menurut Suli & Nirsal (2023), MySQL merupakan basis data yang dibuat menggunakan perangkat lunak sumber terbuka. Hal ini menunjukkan bahwa perangkat lunak tersebut dirancang secara sederhana untuk memenuhi persyaratan sistem dan dapat memungkinkan operasi manajemen data yang adaptif dan efektif.

MySQL merupakan sistem manajemen basis data relasional yang bersifat *open source* dan berfungsi sebagai *server* untuk menyimpan serta

mengambil data sesuai kebutuhan aplikasi, sehingga banyak digunakan dalam berbagai pengembangan sistem.

7. Prototype

Prototype merupakan model awal atau versi pertama dari suatu sistem yang digunakan untuk menggambarkan serta mensimulasikan aplikasi kepada pengguna, sehingga pengembang dapat melakukan penyesuaian sistem sesuai kebutuhan sebelum sistem dikembangkan secara penuh (Syarif & Risdiansyah, 2024).

Metode *Prototype* diawali dengan pembuatan model dasar sistem agar pengembang dan pengguna dapat berinteraksi selama proses pengembangan berlangsung. Melalui metode ini, kebutuhan pengguna dapat dipahami sebelum sistem dibangun secara lengkap (Kurniati, 2021).

Metode *Prototype* digunakan untuk memperoleh masukan dari pengguna terhadap sistem yang dirancang, sehingga pengembang dapat menyesuaikan sistem agar sesuai dengan kebutuhan informasi pengguna (Hendri et al., 2022).

Berdasarkan pendapat tersebut, *Prototype* merupakan metode pengembangan sistem dengan membuat model awal untuk membantu pengembang memahami kebutuhan pengguna sebelum sistem dibangun secara lengkap.

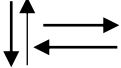
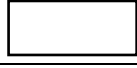
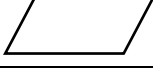
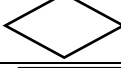
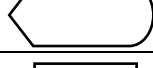
8. Flowchart

Flowchart merupakan diagram yang menggambarkan alur suatu sistem, proses, serta pengendalian yang diterapkan dalam sebuah organisasi atau perusahaan (Tuasamu et al., 2023). “A flowchart is a set of procedures

for resolving an issue that are denoted by unique symbols. The program flow's logic is shown in this flowchart.” Dijelaskan bahwa serangkaian langkah untuk mengatasi suatu masalah yang diwakili oleh simbol-simbol tertentu. Dasar pemikiran di balik alur suatu program ditunjukkan dalam diagram alir ini (Tarsini & Anggraeni, 2024).

Flowchart merupakan diagram yang menggambarkan alur proses, logika, dan tahapan suatu sistem secara sistematis dengan menggunakan simbol tertentu agar mudah dipahami.

Tabel 2. 1 Simbol *Flowchart*

No	Simbol	Nama	Fungsi
1		<i>Flow</i>	Menjelaskan aliran proses dalam sebuah sistem atau program
2		<i>Terminal</i>	Menunjukkan awal dan akhir suatu proses atau program
3		<i>Process</i>	Menyatakan suatu proses
4		<i>Input / Output</i>	<i>Input / Output</i> data
5		<i>Decision</i>	Menyatakan suatu kondisi
6		<i>Internal Storage</i>	Menunjukkan proses penyimpanan data dalam sistem
7		<i>Magnetic Disk</i>	Menunjukkan proses penyimpanan dan pengambilan data pada database
8		<i>Display</i>	Menyatakan alat <i>output</i>
9		<i>Document</i>	<i>Input / Output</i> berbentuk dokumen
10		<i>Manual Operation</i>	Proses manuals`
11		<i>Connector</i>	Penghubung bagian dalam satu halaman
12		<i>Off Page Connector</i>	Penghubung bagian berbeda halaman

Sumber : Tarsini & Anggraeni (2024)

9. Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah bahasa pemodelan standar yang banyak dimanfaatkan dalam bidang rekayasa perangkat lunak untuk mendukung proses perancangan serta dokumentasi sistem berbasis objek. Menurut penjelasan Rosa dan Shalahuddin 2015 dalam Wulandari et al., (2024), *Unified Modeling Language (UML)* merupakan bahasa pemodelan standar yang banyak digunakan dalam rekayasa perangkat lunak berorientasi objek. UML menyediakan cara visual untuk merancang, mendokumentasikan dan memahami sistem perangkat lunak secara menyeluruh.

Menurut Hindarto & Hariadi (2023), mengungkapkan bahwa *"Unified Modeling Language (UML) is a standardized visual modeling language used to design, represent, and document the structure and behavior of software systems."* Dijelaskan bahwa Unified Modeling Language (UML) merupakan bahasa pemodelan visual yang digunakan untuk merancang, menggambarkan, dan mendokumentasikan sistem perangkat lunak.

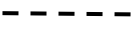
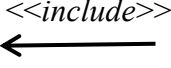
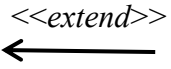
a. Use Case Diagram

Use Case adalah metode dalam analisis dan perancangan sistem yang digunakan untuk menggambarkan interaksi antara pengguna (aktor) dengan sistem dalam mencapai tujuan tertentu. Menurut Wicaksono et al., (2023) mengatakan bahwa *Use Case* diagram merupakan representasi visual yang memperlihatkan hubungan antara aktor dan *use case* serta Batasan sistem yang relevan. Diagram ini membantu mengidentifikasi

fungsionalitas sistem dari perspektif pengguna dan memberikan gambaran umum mengenai cara penggunaan sistem.

Selain itu Harlina et al., (2025), *Use Case* diagram merupakan memodelkan perilaku sistem yang diinginkan dan dibutuhkan pengguna, yang bermanfaat dalam menentukan struktur organisasi. Hal ini menunjukkan *Use Case Diagram* memainkan peran penting dalam menguraikan persyaratan fungsional sistem dan menjelaskan hubungan antara pengguna (aktor) dan fungsi-fungsi yang dapat diakses dalam sistem, sehingga memungkinkan proses desain sistem untuk dijalankan sesuai dengan tuntutan pengguna.

Tabel 2. 2 Simbol *Use Case Diagram*

No	Simbol	Nama	Keterangan
1		Aktor	Mewakili peran orang
2		<i>Use Case</i>	Abstraksi antara sistem dan aktor
3		<i>Association</i>	Penghubung <i>use case</i> dengan aktor
4		Generalisasi	Spesialisasi aktor untuk berhubungan dengan <i>use case</i>
5		<i>Include</i>	Simbol yang menunjukkan bahwa fungsi dari sebuah <i>use case</i> telah terintegrasikan ke dalam <i>use case</i> lain, sehingga selalu dieksekusi sebagai bagian proses tersebut.
6		<i>Extend</i>	Sebuah simbol yang jika suatu kondisi terpenuhi, menunjukkan bahwa sebuah <i>use case</i> merupakan tambahan fungsional untuk <i>use case</i> lainnya.

Sumber : Harlina et al., (2025)

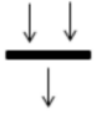
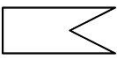
b. Activity Diagram

Activity diagram adalah jenis diagram yang berfungsi untuk memodelkan aliran kerja (*workflow*) dari suatu proses atau kegiatan dalam sistem. Diagram ini menampilkan urutan aktivitas, kondisi, serta pengambilan keputusan yang terjadi selama proses tersebut berlangsung.

Menurut Gedam & Meshram, 2023 , ” *An activity diagram is utilized to illustrate the visual flow of activities or events that occur within a use case diagram.* ” Dijelaskan bahwa *activity diagram* digunakan untuk menggambarkan alur aktivitas atau kejadian yang berlangsung dalam sebuah *use case diagram* secara visual. Diagram ini menunjukkan urutan proses yang terjadi dalam sistem sehingga hubungan antar aktivitas dapat dipahami dengan lebih jelas.

Activity diagram menjelaskan berbagai alur aktivitas dalam sistem yang sedang dibuat, bagaimana setiap alur dimulai, potensi keputusan, dan bagaimana suatu aktivitas berakhir. Hal ini dapat mempermudah proses analisis dan desain sistem dengan memberikan gambaran visual dari setiap tindakan, pilihan, dan prosedur yang terjadi (Suli & Nirsal, 2023).

Tabel 2. 3 Simbol *Activity Diagram*

No	Simbol	Nama	Keterangan
1		<i>Start</i>	Status awal merupakan awal dari aktivitas sistem.
2		<i>Activity</i>	Bertujuan untuk menggambarkan aktivitas dari sebuah sistem
3		<i>Connector</i>	Ini menunjukkan alur arah atau alur kontrol suatu aktivitas.
4		<i>Joint / Sinkronisasi ion bar</i>	Penggabungan digunakan untuk menghubungkan aktivitas satu dengan aktivitas yang lain.
5		<i>Fork</i>	Dua operasi bersamaan dipisahkan dari satu aliran aktivitas utama.
6		<i>Decision</i>	<i>Decision</i> / percabangan merupakan aktivitas yang memiliki lebih dari satu kondisi atau pilihan.
7		<i>Send Signal</i>	Menyampaikan ke aktivitas penerima bahwa suatu sinyal sedang dikirim.
8		<i>Receive Signal</i>	Menunjukkan bahwa suatu peristiwa telah diterima. Alur yang dihasilkan dari tindakan ini selesai setelah peristiwa diterima.
9		<i>Option Loop</i>	Hal ini memberi desainer kemampuan untuk menggambarkan urutan berulang didalam simbol <i>loop</i> .
10		<i>Flow final</i>	Menandakan akhir dari suatu aliran proses tertentu. akhir suatu proses harus ditandai dengan simbol akhir aliran.
11		<i>End</i>	Menandakan akhir suatu aktivitas dan akhir semua aliran proses.

Sumber: Gedam & Meshram (2023)

c. *Sequence Diagram*

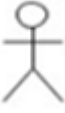
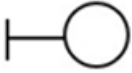
Sequence diagram adalah urutan pesan atau panggilan objek antar objek dalam suatu sistem, selain itu juga menggambarkan bagaimana item-item ini berinteraksi pada berbagai titik waktu. Menurut Gunawan et al., (2023) menjelaskan bahwa *Sequence diagram* adalah salah satu jenis diagram UML yang menunjukkan interaksi berurutan antar item dalam

suatu sistem. Biasanya, grafik ini digunakan untuk menunjukkan bagaimana suatu fitur atau fungsi dalam suatu sistem masuk akal.

Menurut Al-Fedaghi (2021) mengatakan bahwa *“the second most popular UML diagram that depicts how objects interact and exchange messages over time is the sequence diagram, a behavioral representation.”* Dijelaskan bahwa *sequence diagram* digunakan untuk menggambarkan interaksi antar objek melalui pertukaran pesan secara berurutan, sehingga memudahkan pemahaman terhadap alur proses dalam suatu sistem.

Menurut Hindarto & Hariadi (2023) menjelaskan bahwa *“A sequence diagram represents the chronological sequence of messages exchanged between system objects, illustrating how different components interact to perform a specific process.”* Dijelaskan bahwa *sequence diagram* merupakan diagram yang digunakan untuk menggambarkan urutan interaksi atau pertukaran pesan antara objek-objek dalam suatu sistem secara kronologis. Diagram ini menunjukkan bagaimana setiap komponen dalam sistem saling berkomunikasi dan berinteraksi untuk menjalankan suatu proses tertentu.

Tabel 2. 4 Simbol *Sequence Diagram*

Simbol	Elemen	Keterangan
	<i>Actor</i>	Menggambar ilustrasi seseorang yang menggunakan suatu sistem.
	<i>Entity Class</i>	Menggambarkan hubungan yang akan terjadi.
	<i>Boundary Class</i>	Mengelola pertukaran informasi antar lingkungan sistem.
	<i>Control Class</i>	Menjelaskan hubungan antara <i>boundary</i> dan tabel.
	<i>A Focus of Control & A Life Line</i>	Menjelaskan titik awal dan akhir <i>message</i> tersebut.
	<i>A Message</i>	Menjelaskan cara mengirim pesan.

Sumber: Gunawan et al., (2023)

d. *Class Diagram*

Class diagram adalah sistem berorientasi objek dapat dibangun dan ditampilkan menggunakan notasi grafis. Dalam *Unified Modeling Language* (UML), *class diagram* adalah jenis diagram struktur statis yang menjelaskan struktur suatu sistem.

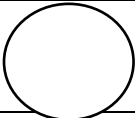
Menurut Suli & Nirsal (2023) menjelaskan bahwa *class diagram* merupakan Jenis diagram yang paling populer adalah representasi visual dari kelas-kelas suatu sistem. Diagram ini menggambarkan hubungan antar kelas dan memberikan deskripsi menyeluruh tentang setiap kelas dalam model desain sistem.

Class diagram merupakan salah satu jenis diagram struktural UML yang memberikan deskripsi jelas tentang kelas, atribut, metode, dan hubungan setiap objek statis, dengan kata lain diagram kelas menjelaskan

hubungan yang ada, bukan apa yang terjadi jika kelas-kelas tersebut saling terkait (Sabila & Lubis, 2025).

Berdasarkan berbagai sudut pandang yang telah dijelaskan diatas, dapat disimpulkan bahwa *Class Diagram* adalah representasi struktural statis dalam UML yang berperan dalam pemodelan sistem berorientasi objek. Sebagai dasar desain sistem, diagram ini menunjukkan kelas-kelas beserta properti, metode, dan interaksi antar kelas.

Tabel 2. 5 Simbol *Class Diagram*

No.	Simbol	Nama	Keterangan
1.		<i>Class</i>	Menunjukkan kelas yang menjadi bagian penyusun suatu sistem
2.		<i>Interface</i>	Memiliki konsep yang sama seperti <i>interface</i> pada pemrograman berorientasi objek.
3.		<i>Association</i>	Menggambarkan hubungan umum antar kelas dan umumnya dapat disertai dengan <i>Multiplicity</i> .
4.		<i>Direct Association</i>	Menunjukkan hubungan antar kelas yang mengarah pada penggunaan suatu kelas oleh kelas lainnya, serta biasanya dilengkapi dengan <i>Multiplicity</i> .
5.		Generalisasi	Menunjukkan hubungan antara kelas induk (umum) dan kelas turunan (khusus).
6.		<i>Dependency</i>	Menggambarkan adanya hubungan ketergantungan antara satu kelas dengan kelas lainnya.
7.		<i>Aggregation</i>	Menunjukkan hubungan keseluruhan dan bagian (<i>whole-part</i>) antar kelas.

Sumber : (Harlina et al., 2025)

B. Kajian Empiris

Penelitian ini dalam penyusunannya menggunakan beberapa referensi dari jurnal dan penelitian terdahulu yang relevan dengan topik yang diangkat,

yaitu Rancang Bangun Sistem Informasi Absensi Dan Penggajian Berbasis *Website* Di Mie Tjahaya Madiun. Referensi tersebut digunakan sebagai acuan dalam perancangan dan pengembangan sistem yang akan dibangun. Studi literatur dari penelitian sebelumnya dimanfaatkan untuk memperkuat landasan penelitian serta membantu dalam menganalisis, membandingkan, dan mengevaluasi sistem yang telah dikembangkan sebelumnya, sehingga dapat menghasilkan sistem yang lebih efektif dan sesuai dengan kebutuhan.

Maulana et al., (2024) melakukan penelitian mengenai analisis dan perancangan sistem informasi absensi dan penggajian berbasis *website*. Hasil penelitian menunjukkan bahwa penerapan sistem tersebut dapat menyederhanakan proses pencatatan kehadiran dan perhitungan gaji secara otomatis, sehingga pekerjaan menjadi lebih efisien serta mengurangi risiko kesalahan dalam penggajian. Persamaan penelitian ini terletak pada penggunaan sistem berbasis *website* untuk mengelola absensi dan penggajian, sedangkan perbedaannya terletak pada objek penelitian dan fitur tambahan yang digunakan.

Selanjutnya, penelitian oleh Shofi et al. (2021) mengembangkan sistem absensi karyawan berbasis online dengan memanfaatkan teknologi. Hasil penelitian menunjukkan bahwa sistem mampu meningkatkan efisiensi proses absensi serta meminimalisir kecurangan dalam pencatatan kehadiran. Persamaan penelitian ini terletak pada pengembangan sistem absensi berbasis *digital*, sedangkan perbedaannya adalah metode validasi yang digunakan, di mana penelitian ini menggunakan *barcode*, sedangkan penelitian yang dilakukan menggunakan GPS.

Penelitian lain oleh Effendi et al. (2023) membahas tentang sistem informasi berbasis *website* dalam pengelolaan data organisasi. Hasil penelitian membuktikan bahwa implementasi sistem informasi berbasis *website* dapat mendukung pengelolaan data secara lebih optimal serta memberikan kemudahan dalam memperoleh informasi secara cepat dan tepat. Persamaan penelitian ini terletak pada penggunaan sistem berbasis *website*, sedangkan perbedaannya terletak pada fokus sistem yang dikembangkan.

Selain itu, penelitian yang dilakukan oleh Dinka et al., (2022) menjelaskan bahwa keberadaan sistem informasi memberikan kontribusi penting bagi organisasi, khususnya dalam mendukung aktivitas pengelolaan data perusahaan. Pemanfaatan sistem informasi dalam proses bisnis mampu membantu pengolahan data menjadi informasi yang lebih terstruktur sehingga dapat mendukung kegiatan operasional secara efektif. Hal ini menunjukkan bahwa implementasi sistem informasi yang tepat dapat meningkatkan kinerja perusahaan serta membantu proses pengambilan keputusan berdasarkan informasi yang lebih akurat.

Berdasarkan tinjauan penelitian sebelumnya, sistem informasi kehadiran dan penggajian berbasis *website* terbukti mendukung pengelolaan data karyawan yang lebih akurat dan efisien. Berdasarkan temuan ini, studi ini menciptakan sistem yang mencakup notifikasi waktu nyata dan validasi posisi berbasis GPS untuk meningkatkan produk akhir dan memberikan nilai tambah dibandingkan studi sebelumnya.

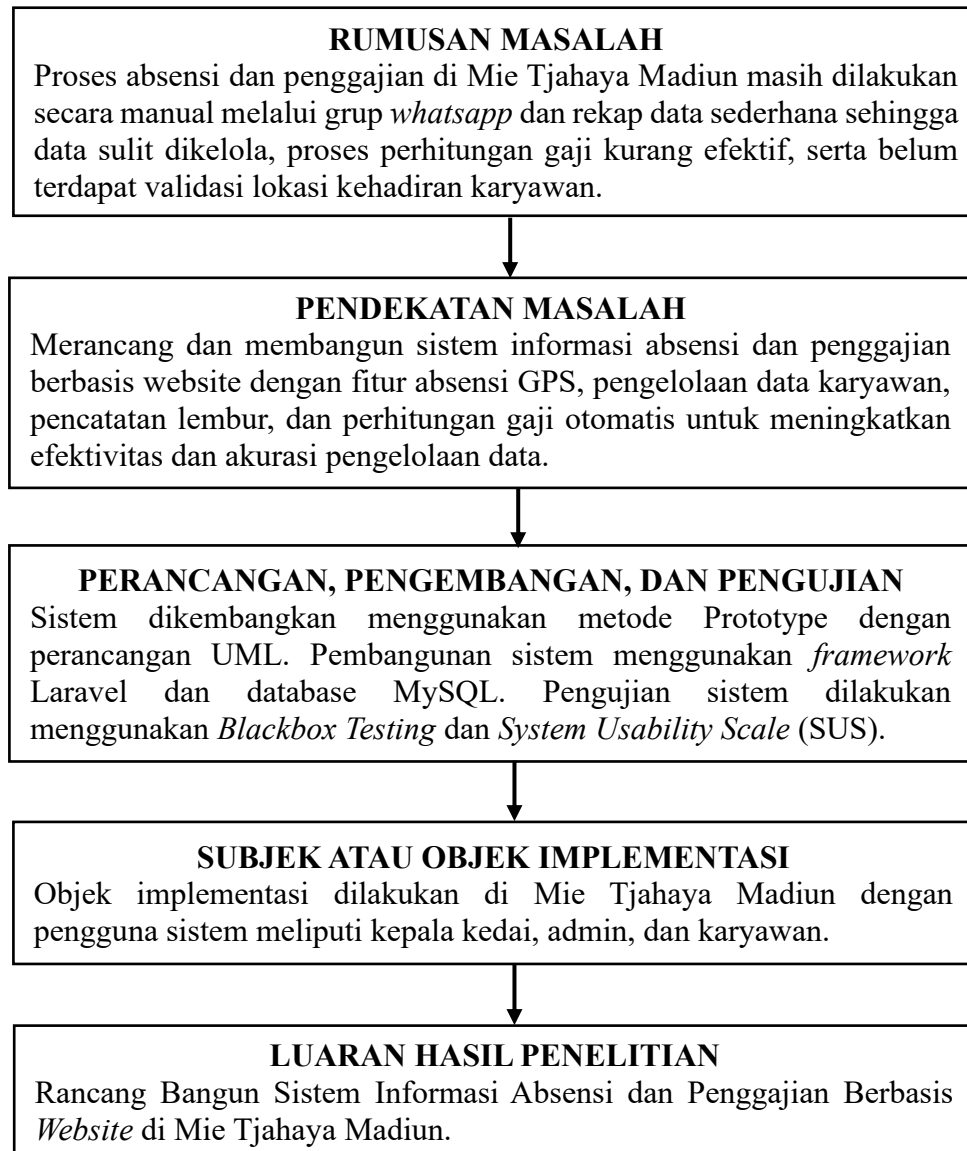
C. Kerangka Berfikir

Permasalahan utama yang mendasari penelitian ini adalah proses absensi dan penggajian di Mie Tjahaya Madiun yang masih dilakukan secara manual melalui grup *WhatsApp* dan rekap data sederhana. Kondisi tersebut menyebabkan data absensi sulit dikelola, proses perhitungan gaji memerlukan waktu yang lama, serta berpotensi menimbulkan kesalahan dalam pengolahan data. Selain itu, sistem yang digunakan belum memiliki validasi lokasi sehingga pengawasan kedisiplinan karyawan menjadi kurang efektif.

Sebagai upaya mengatasi permasalahan tersebut, dibutuhkan sistem informasi berbasis *website* yang mengintegrasikan pengelolaan data absensi dan penggajian. Dengan adanya sistem tersebut, proses pengolahan data dapat berlangsung secara lebih efektif, tepat serta meminimalkan kesalahan dalam pengelolaan informasi. Selain itu, sistem dilengkapi dengan fitur absensi berbasis GPS, pengelolaan data karyawan, pencatatan lembur, serta perhitungan gaji otomatis berdasarkan data kehadiran dan keterlambatan.

Dalam merancang dan mengembangkan sistem informasi ini, penelitian menggunakan metode *Prototype*. Metode *Prototype* dipilih karena memfasilitasi komunikasi langsung antara pengembang dan pengguna melalui rancangan awal sistem sehingga kebutuhan pengguna dapat diketahui dengan lebih jelas. Perancangan sistem dilakukan menggunakan UML, sedangkan pengujian sistem menggunakan metode *Blackbox Testing* dan *System Usability Scale* (SUS). Dengan penerapan metode tersebut, sistem yang dikembangkan diharapkan dapat sesuai dengan kebutuhan operasional Mie Tjahaya Madiun. Berikut ini

merupakan bagan kerangka berpikir yang menggambarkan alur penelitian dari identifikasi masalah hingga luaran hasil penelitian.



Gambar 2. 1 Kerangka Berpikir

